



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Kadir (2017:2), Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia. Meskipun komputer berasal dari kata “Komputasi”, komputasi yang memang dilaksanakannya mungkin tidak terlihat secara eksplisit. Ketika orang menggunakan komputer untuk membuat dokumen, berbagai perhitungan yang dilakukan tidak terlihat. Sebagai contoh, ketika pemakai memilih pengaturan “*justify*” atau rata di batas kiri dan batas kanan dokumen, perhitungan untuk menambahkan karakter-karakter spasi sebenarnya terjadi.

Suryana (2014:2), Komputer terdiri atas perangkat keras dan perangkat lunak salah satu elemen perangkat keras ialah komputer. Komputer akan digunakan sebagai sarana untuk mengendalikan sistem informasi yang berbasis komputer, yang merupakan salah satu bentuk aplikasi dari teknologi informasi. Dengan elemen – elemen perangkat keras komputer *input*, *process* dan *output*.

Dari definisi diatas penulis menyimpulkan bahwa komputer adalah merupakan perangkat keras yang dapat digunakan untuk membantu dalam berbagai pekerjaan manusia seperti pembuatan dokumen.

2.1.2 Pengertian Perangkat Lunak

Kadir (2017:2), Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada *computer* agar dapat melaksanakan tugas sesuai kehendak pemakai. System operasi seperti *Windows*, *Mac OS*, dan *Linux*, dan aplikasi seperti *Microsoft Word* dan *Microsoft Excel* adalah contoh perangkat lunak.

Dari definisi diatas penulis menyimpulkan bahwa perangkat lunak adalah suatu program aplikasi yang digunakan sesuai dengan perintah yang ditujukan untuk menjalankan tugasnya.



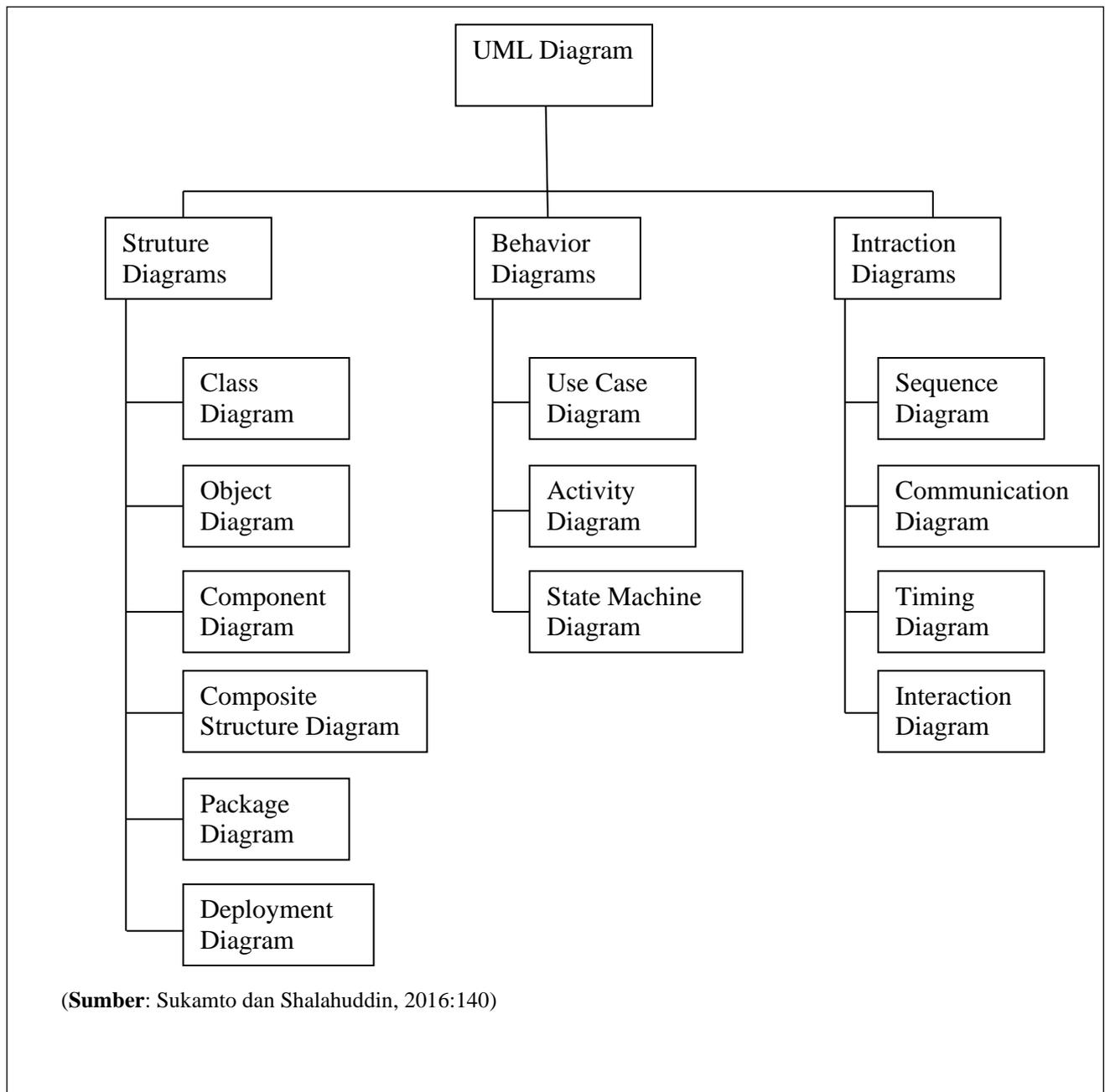
2.2 Teori Khusus

2.2.1 Pengertian UML (*Unified Modeling Language*)

Sukanto dan Shalahuddin (2016:133), UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

2.2.2 Macam - macam diagram *Unified Modeling Language* (UML)

Sukanto dan Shalahuddin (2016:140), Pada *UML* terdapat 13 buah diagram yang selanjutnya dikelompokkan menjadi 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



Gambar 2.1 Macam - macam diagram *Unified Modeling Language* (UML)

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

- a. *Structure Diagram*



Adalah sekumpulan diagram yang digunakan untuk menggambarkan struktur statis dari sebuah sistem yang dimodelkan.

b. *Behavior Diagram*

Adalah sekumpulan diagram yang digunakan untuk menggambarkan rangkaian perubahan yang terjadi pada sebuah sistem.

c. *Interaction Diagram*

Adalah sekumpulan diagram yang digunakan untuk menggambarkan sebuah interaksi sistem dengan sistem lain nya ataupun interaksi antar subsistem pada suatu sistem.

2.2.3 Pengertian *Class Diagram*

Sukanto dan Shalahuddin (2016:141), Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- a. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas
- b. Operasi atau model adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Tabel 2.1 Simbol-simbol dalam *class diagram*

Simbol	Deskripsi
Kelas <div style="border: 1px solid black; padding: 5px; width: fit-content;"> Nama_kelas +atribut +operasi() </div>	Kelas pada struktur system
Antarmuka/ <i>interface</i> <div style="text-align: center;">○</div> Nama_ <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek

Lanjutan Tabel 2.1 Simbol-simbol dalam *class diagram*



Simbol	Deskripsi
Asosiasi/ <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah/ <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Keberganungan/ <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi/ <i>aggregation</i> 	Relasi antarkelass dengan makna semua-bagian (<i>whole-part</i>)

(Sumber: Sukamto dan Shalahuddin, 2016:146-147)

2.2.4 Pengertian *Use Case Diagram*

Sukamto dan Shalahuddin (2016:155), *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

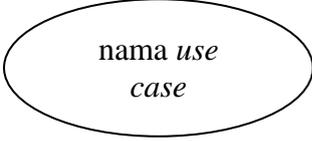
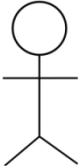
- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu



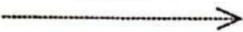
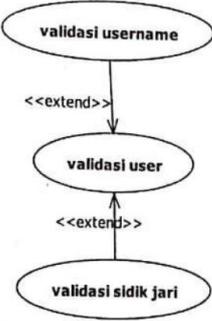
sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Tabel 2.2 Simbol-simbol dalam *Use case*

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal di awal <i>frase</i> nama <i>use case</i>.</p>
<p>Aktor/<i>actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal <i>frase</i> nama aktor.</p>
<p>Asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>

Lanjutan Tabel 2.2 Simbol-simbol dalam *Use case*

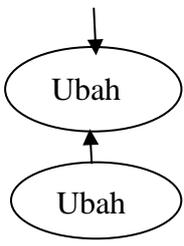
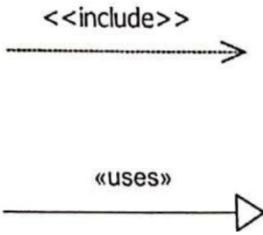
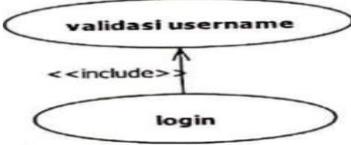
Simbol	Deskripsi
<p>Ekstensi/<i>extend</i></p> <p style="text-align: center;"><<extend>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal;</p>  <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>

Lanjutan Tabel 2.2 Simbol-simbol dalam *Use case*

<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>
---	---

Ubah



	 <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan/<i>include/uses</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di use case:</p> <ol style="list-style-type: none"> 1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, missal pada kasus berikut:  2. <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan selalu dijalankan sebelum <i>use case</i> tambahan



	<p>dijalankan, misal pada kasus berikut:</p> <p>Kedua interpretasi di atas dapat dianut salah satu keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>
--	--

Sumber: Sukamto dan Shalahuddin, 2016:156-158)

2.2.5 Pengertian *Activity Diagram*

Sukamto dan Shalahuddin (2016:161), *Activity Diagram* atau Diagram Aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
4. Rancangan menu yang ditampilkan pada perangkat lunak

Tabel 2.3 Simbol-simbol dalam *Activity Diagram*

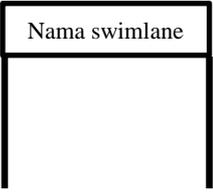
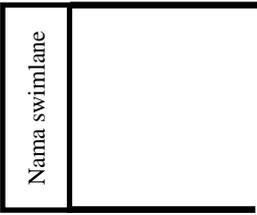
Simbol	Deskripsi
Status awal	Status awal aktivitas sistem, sebuah



	diagram aktivitas memiliki sebuah status awal
<p data-bbox="316 421 438 454">Aktivitas</p> 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
<p data-bbox="316 696 619 730">Percabangan / <i>decision</i></p> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
<p data-bbox="316 972 587 1005">Penggabungan / <i>join</i></p> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
<p data-bbox="316 1140 475 1173">Status akhir</p> 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir



Lanjutan Tabel 2.3 Simbol-simbol dalam Activity Diagram

<p>Swimlane</p>  <p>Atau</p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>
--	--

(Sumber: Sukamto dan Shalahuddin, 2016:162-163)

2.2.6 Pengertian Sequence Diagram

Sukamto dan Shalahuddin (2016:165), Diagram sekuen atau *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstantiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use*

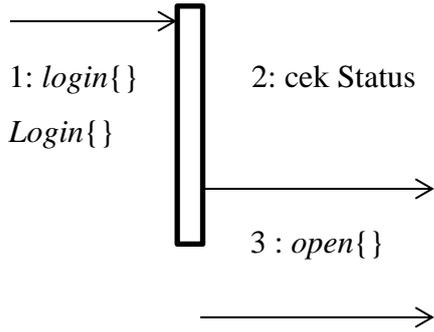


case yang telah didefinisikan interaksinya jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

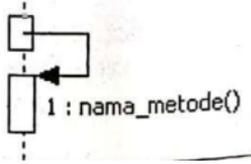
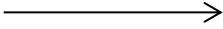
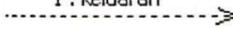
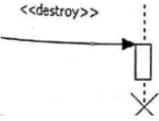
Tabel 2.4 Simbol-simbol dalam *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p>  <p>Nama aktor</p> <p>Atau</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Namaaktor</div> <p>tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama <i>actor</i></p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <u>Nama objek</u> <u>:nama kelas</u> </div>	<p>Menyatakan objek yang berinteraksi pesan</p>

Lanjutan Tabel 2.4 Simbol-simbol dalam *Sequence Diagram*

<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misalnya</p>  <p>Maka cek Status <i>Login{ }</i> dan <i>open{ }</i> dilakukan di dalam metode <i>login{ }</i></p> <p>Aktor tidak memiliki waktu aktif</p>
<p>Pesan tipe <i>create</i></p> <p><code><<create>></code></p> 	<p>Menyatakan suatu objek membuat objek lain, arah panah mengarah pada arah objek yang dibuat</p>
<p>Pesan tipe <i>call</i></p> <p>1 :nama_metode{ }</p>	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri</p>

Lanjutan Tabel 2.4 Simbol-simbol dalam *Sequence Diagram*

	 <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
<p>Pesan tipe <i>send</i></p> <p>1 : masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe <i>return</i></p> <p>1 : keluaran</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i></p>

(Sumber: Sukanto dan Shalahuddin, 2016:165-167)



2.3 Teori Judul

2.3.1 Pengertian Aplikasi

Menurut Koyuko dkk dikutip Oki (2016), Aplikasi adalah suatu *software* yang dapat melakukan beberapa tugas tertentu seperti membuat dokumen, penambahan data dan beberapa tugas lainnya. *suite* aplikasi (*application suite*) adalah penggabungan dari berbagai macam aplikasi menjadi satu. salah satu contohnya *Microsoft Office* dan *Open Office* yaitu menggabungkan aplikasi *Microsoft word*, *Microsoft excel*, dan beberapa aplikasi lainnya. Pada umumnya aplikasi-aplikasi yang berada dalam suatu paket mempunyai *interface* yang sama sehingga memudahkan *user* untuk mempelajari dan memakai aplikasi tersebut dan juga aplikasi-aplikasi tersebut mempunyai bisa berinteraksi antar aplikasi sehingga memudahkan *user*. Contohnya, suatu lembar kerja dapat disimpan dalam suatu dokumen pengolah kata meskipun dibuat pada aplikasi lembar kerja yang terpisah.

2.3.2 Pengertian Penerimaan

Menurut Kamus Besar Bahasa Indonesia dalam situsnya <https://kbbi.kemdikbud.go.id/entri/penerimaan>, Penerimaan adalah proses, cara, perbuatan menerima; penyambutan.

2.3.3 Pengertian Web

Abdulloh (2018:1), *Website* dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia. Halaman *website* dibuat menggunakan bahasa standard yaitu HTML. Skrip HTML ini akan diterjemahkan oleh *web browser* sehingga dapat ditampilkan dalam bentuk informasi yang dapat dibaca oleh semua orang.

2.3.4 Pengertian Aplikasi Penerimaan Peserta Didik Baru (PPDB) Pada Yayasan Pondok Pesantren Ma'ariful Ulum Berbasis Website

Dimana sistem ini digunakan untuk melakukan penerimaan peserta didik baru. Pada penelitian ini akan dibangun aplikasi yang dapat digunakan untuk



memudahkan peserta didik baru dalam pendaftaran dan menerima pemberitahuan tentang diterima atau tidaknya pendaftaran yang dilakukan. Aplikasi yang dibuat haruslah dapat diakses melalui web sehingga dapat diakses secara langsung oleh peserta ketika terhubung dengan Internet.

2.4 Teori Program

2.4.1 Pengertian Basis Data

Fathansyah (2018:2), Basis Data terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Sebagai satu kesatuan istilah, Basis Data (*Database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

2.4.2 Pengertian MySQL

Menurut Wahana Komputer (2010:21) dalam firman (2016:30), MySQL adalah database server open source yang cukup populer keberadaannya. Dengan berbagai keunggulan yang dimiliki, membuat software database ini banyak digunakan oleh praktisi untuk membangun suatu project. Adanya fasilitas API (Application Programming Interface yang dimiliki oleh MySQL, memungkinkan bermacam-macam aplikasi komputer yang ditulis dengan berbagai bahasa pemrograman dapat mengakses basis data MySQL.

2.4.3 Pengertian Bahasa Pemrograman

Menurut Wahyudi dkk (2013:65), Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer atau bahasa pemrograman komputer,



adalah instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

Menurut tingkat kedekatannya dengan mesin komputer, bahasa pemrograman terdiri dari:

1. Bahasa Mesin, yaitu memberikan perintah kepada komputer dengan memakai kode bahasa biner, contohnya 01100101100110
2. Bahasa Tingkat Rendah, atau dikenal dengan istilah bahasa rakitan (bah. Inggris Assembly), yaitu memberikan perintah kepada komputer dengan memakai kode-kode singkat (kode mnemonic), contohnya MOV, SUB, CMP, JMP, JGE, JL, LOOP, dsb.
3. Bahasa Tingkat Menengah, yaitu bahasa komputer yang memakai campuran instruksi dalam kata-kata bahasa manusia (lihat contoh Bahasa Tingkat Tinggi di bawah) dan instruksi yang bersifat simbolik, contohnya {, }, ?, <>, &&, ||, dsb.
4. Bahasa Tingkat Tinggi, yaitu bahasa komputer yang memakai instruksi berasal dari unsur katakata bahasa manusia, contohnya begin, end, if, for, while, and, or, dsb.

2.4.4 Pengertian HTML

Abdulloh (2018:7), *HTML* merupakan singkatan dari *Hypertext Markup Language* yaitu bahasa standar *web* yang dikelola penggunaannya oleh W3C (*World Wide Web Consortium*) berupa tag-tag yang menyusun setiap elemen dari *website*. *HTML* berperan sebagai penyusun struktur halaman *website* yang menempatkan setiap elemen *website* sesuai layout yang diinginkan.

HTML biasanya disimpan dalam sebuah file berekstensi *.html*. untuk mengetikkan skrip *HTML* dapat menggunakan *text editor* seperti Notepad sebagai bentuk paling sederhana atau *text editor* khusus yang dapat mengenali setiap unsur



skrip *HTML* dan menampilkannya dengan warna yang berbeda sehingga mudah dibaca, seperti Notepad++. *Sublime Text*, dan masih banyak lagi aplikasi lain yang sejenis.

2.4.5 Pengertian PHP

Menurut Kustiyaningsih (2011:114) dalam firman (2016:30), “PHP (atau resminya PHP: Hypertext Preprocessor) adalah skrip bersifat server-side yang ditambahkan ke dalam HTML”. Pada prinsipnya server akan bekerja apabila ada permintaan dari client. Dalam hal ini client menggunakan kode-kode PHP untuk mengirimkan permintaan ke server. Sistem kerja dari PHP diawali dengan permintaan yang berasal dari halaman website oleh browser. Berdasarkan URL atau alamat website dalam jaringan internet, browser akan menemukan sebuah alamat dari webserver, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh webserver.

Selanjutnya webserver akan mencari berkas yang diminta dan menampilkan isinya di browser. Browser yang mendapatkan isinya segera menerjemahkan kode HTML dan menampilkannya. Lalu bagaimana apabila yang dipanggil oleh user adalah halaman yang mengandung script PHP? Pada prinsipnya sama dengan memanggil kode HTML, namun pada saat permintaan dikirim ke web-server, web-server akan memeriksa tipe file yang diminta user. Jika tipe file yang diminta adalah PHP, maka akan memeriksa isi script dari halaman PHP tersebut.

Apabila dalam file tersebut tidak mengandung script PHP, permintaan user akan langsung ditampilkan ke browser, namun jika dalam file tersebut mengandung script PHP, maka proses akan dilanjutkan ke modul PHP sebagai mesin yang menerjemahkan script-script PHP dan mengolah script tersebut, sehingga dapat dikonversikan ke kode-kode HTML lalu ditampilkan ke browser user.

2.4.6 Script PHP

Menurut Hidayatullah dan Kawistara (2017:231), sintaks dasar PHP adalah sebagai berikut:



1. PHP Opening dan Closing Tag

```
<?php /* PHP tag pembuka */  
  
/*  
    ISI PROGRAM PHP  
*/  
  
/* PHP tag penutup */  
?>
```

2. Komentar dalam PHP

```
<?php  
echo 'Ini adalah contoh'; //contoh gaya komentar satu baris  
  
/* Ini adalah contoh komentar  
lebih dari satu baris */  
  
echo 'Contoh lagi'; #Contoh gaya komentar satu baris pada shell  
?>
```

2.4.7 Pengertian Web Server

Menurut Fathansyah (2012:466) dalam prayitno & safitri (2015:2), Menerangkan bahwa pengertian web server adalah “Server Web (Web Server) merujuk pada perangkat keras (server) dan perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP ataupun variannya (seperti FTP dan HTTPS) atas berkas-berkas yang terdapat pada suatu URL ke pemakai”.



2.4.8 Pengertian Xampp

Menurut Wahana(2009:30) dalam prayitno & safitri (2015:2), “XAMPP adalah salah satu paket instalasi apache, PHP, dan MySQL secara instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut”.

2.4.9 Pengertian phpMyAdmin

Nugroho (2014:10), phpMyAdmin adalah aplikasi manajemen *database server* MySQL berbasis *web*. Dengan aplikasi phpMyAdmin kita bias mengelola *database* sebagai **root** (pemilik server) juga disebut *Administrator database*. Lewat fasilitas phpMyAdmin, Anda bias membuat *database* baru, *table* dan struktur data mengelola data di dalam *database*.